# How to dri

## How to DRI: Command Line Tool

There are a number of ways to ingest or deposit content into the DRI Repository. One method is ingest via the DRI client tool. This is a command-line application providing access via an API to the DRI Repository. This guide describes the software prerequisites and installation procedure for the DRI client. The command line tool outputs JSON by default, making it easy to call it from within other applications.

A list of all the command line arguments and their functionality are contained in the tools **README.rdoc** file which also contains an example workflow.

If you are using this tool to bulk ingest digital objects please review the naming conventions described below to inform any pre-ingest preparation work.

### Prerequisites

The command-line tool is written for Unix or Mac OSX operating systems, it has not been tested under Windows. The command-line tool is developed in the Ruby programming language which is an interpreted language, and thus you will need Ruby installed on your machine in order to run the client tool. The application is tested with Ruby version 2.0.0 but should work with other recent Ruby versions.

The following Ruby gems are also required:

- gli version 2.5.4
- rest-client
- nokogiri
- mimemagic

These gems will be installed automatically as part of the installation process described below.

### Installation

The application can be found in the DRI git repository at git@tracker.dri.ie:drirepo/dri-cli.git

Check out the repository with

```
$ git clone git@tracker.dri.ie:drirepo/dri-cli.git
```

Then run bundle to install the necessary environment

```
$ cd dri

$ bundle install
```

To run the client tool, type

```
$ bundle exec dri [global options] command [command options] [arguments...]
```

## Naming Conventions and Folder Set-Up

Data for ingestion should be arranged with valid descriptive metadata in one folder and the assets using the same naming convention in another. The tool can then be passed the parent directory as an argument and should pick up the metadata and corresponding asset(s) automatically. For example,

```
mydir
 |- data
 |- metadata
```

Metadata and data files for a given object should have the same base filename, for example, the metadata file myobject_1.xml corresponds to the asset file myobject_1.pdf.

If you have more than one asset file for a given object then an underscore and sequence number should be appended to the basename, for example:

- myobject_1_001.tiff
- myobject_1_002.tiff
- myobject_1_003.pdf

If the file extensions are different you could alternatively use the same basename without the sequence number, for example:

- myobject_1.pdf
- myobject_1.tiff

Copy all of your metadata files to the metadata subdirectory (metadata files should have the extension.xml).

Copy your data files to the data subdirectory.

Because the bulk import may take some time to complete it contains a failure recovery mechanism which will allow the user to restart the process should it fail.

Up to date instructions for using the tool are included in the **README.rdoc** in the tool's root directory and by using the command;

```
bundle exec bin/dri --help
```

## Deposit is underpinned by DRI's Legal Framework

Users can only deposit into, and manage, a collection for which they have explicit access and deposit permissions. Users who deposit content into DRI using the command line tool, here described, are bound by DRI's **Deposit Terms and Conditions.** These should be reviewed before any ingestion into the Repository takes place.

**This document is part of DRI's operational documentation, as such it may change from time to time as features develop. The most recent version will always be published on our repository website and these documents should be consulted in conjunction with our operational documentation as necessary.**

**Visit www.dri.ie for more information about the DRI project and our repository.**